



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

**ROZŠIŘUJÍCÍ PERIFERIE PRO MIKROKONTROLÉRY
MC9S08LH**

EXPANSION PERIPHERALS FOR MC9S08LH MICROCONTROLLERS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Adan Nideev

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Macho, Ph.D.

BRNO 2019

Bakalářská práce

bakalářský studijní obor **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

Student: Adan Nideev

ID: 164348

Ročník: 3

Akademický rok: 2018/19

NÁZEV TÉMATU:

Rozšiřující periferie pro mikrokontroléry MC9S08LH

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s mikrokontroléry řady MC9S08LH a vývojovou deskou TWR-S08LH64.
2. Navrhněte připojení následujících rozšiřujících periférií k vývojové desce TWR-S08LH64: rotační enkodér, maticová klávesnice, LCD displej a kalendářový obvod.
3. Navrhněte a realizujte potřebné doplňkové obvody.
4. Navrhněte, naprogramujte a odlaďte alespoň 2 laboratorní úlohy demonstrující použití výše uvedených periférií.
5. Vytvořte srozumitelné návody pro studenty popisující softwarové ovládání uvedených periférií.

DOPORUČENÁ LITERATURA:

MC9S08LH64 Reference Manual. Rev. 5.1. Freescale Semiconductor. 2012.

Termín zadání: 4.2.2019

Termín odevzdání: 20.5.2019

Vedoucí práce: Ing. Tomáš Macho, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Cílem bakalářské práce je seznámení s mikrokontrolérem MC9S08LH64 a deskou TWR-S08LH64. Práce je také zaměřena na připojení následujících rozšiřujících periférií k vývojové desce TWR-S08LH64: rotační enkodér, maticová klávesnice, LCD displej a kalendářový obvod.

Klíčová slova

MC9S08LH64, TWR-S08LH64, IIC, DS3231, LCD displej, F-KV16KEY, enkodér

Abstract

The aim of this bachelor's thesis is to get acquainted with microcontroller MC9S08LH64 and development board TWR-S08LH64. The thesis is also focused on connecting of external peripherals such as rotary encoder, matrix keyboard, LCD display and real-time clock to development board TWR-S08LH64.

Keywords

MC9S08LH64, TWR-S08LH64, IIC, DS3231, LCD display, F-KV16KEY, encoder

Bibliografická citace:

NIDEEV, Adan. *Rozšiřující periferie pro mikrokontroléry MC9S08LH* [online]. Brno, 2019 [cit. 2019-05-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119305>.
Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Tomáš Macho.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma *Rozšiřující periferie pro mikrokontroléry MC9S08LH* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2019**

.....
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Tomáš Macho, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **20. května 2019**

.....
podpis autora

OBSAH

1.	Úvod	11
2.	Mikrokontrolér MC9S08LH64 a vývojová deska TWR-S08LH64.....	12
2.1	Mikrokontroler MC9S08LH64	12
2.2	Vývojová deska TWR-S08LH64	12
3.	IIC sběrnice.....	15
3.1.1	Start signál	15
3.1.2	Odesílání adresy Slave zařízení	16
3.1.3	Přenos dat	16
3.1.4	Stop signál	16
3.2	Modul IIC mikrokontroléru MC9S08LH64	17
3.2.1	IICA (IIC Address Register).....	17
3.2.2	IICF (IIC Frequency Divider Register).....	17
3.2.3	IICC1 (IIC Control Register).....	18
3.2.4	IICS (IIC Status Register)	19
3.2.5	IICD (IIC Data I/O Register).....	20
3.2.6	IICC2 (IIC Control Register 2).....	20
4.	Externí periferie použité pro realizaci laboratorních úloh.....	21
4.1.1	Inkrementální rotační enkodér	21
4.1.2	Maticová klávesnice F-KV16KEY	23
4.1.3	LCD displej 20x04 s IIC převodníkem	24
4.1.4	Modul ZS-042 s kalendářovým obvodem D3231	27
5.	Přípravek pro Připojení periférií	30
5.1.1	Připojení přípravku k desce	32
6.	software pro ovládání periférií.....	35
7.	Návrh laboratorních úloh.....	43
7.1	Laboratorní úloha č.1	43
7.2	Laboratorní úloha č. 2	44
8.	Závěr.....	47
9.	Literatura	48
10.	Seznam zkratek	50

11. Seznam příloh	51
-------------------------	----

Seznam obrázků

Obr. 1 Vývojová deska TWR-S08LH64 [1]	13
Obr. 2 Freescale Tower Systém [2]	13
Obr. 3 Příklad zapojení zařízení při použití IIC sběrnice [4]	15
Obr. 4 Komunikace po IIC sběrnici [3]	15
Obr. 5 Komunikace po IIC sběrnici s Repeated Start signálem [3]	17
Obr. 6 IICA registr [3]	17
Obr. 7 IICF registr [3]	18
Obr. 8 IICC1 registr [3]	18
Obr. 9 IICS registr [3]	19
Obr. 10 IICD registr [3]	20
Obr. 11 IICC2 registr [3]	20
Obr. 12 Enkodér [7]	21
Obr. 13 Průběh signálů enkodéru [7]	22
Obr. 14 Modul KY-040	22
Obr. 15 Maticová klávesnice F-KV16KEY [8]	23
Obr. 16 Maticové zapojení klávesnice [9]	23
Obr. 17 LCD displej 2004a [10]	24
Obr. 18 Umístění vývodů LCD displeje	24
Obr. 19 Připojení IIC převodníku k LCD displeji [6]	25
Obr. 20 Piny A0, A1, A2 IIC převodníku [6]	25
Obr. 21 Časový diagram LCD displeje [5]	27
Obr. 22 Modul ZS-042 [11]	27
Obr. 23 Schéma zapojení ZS-042 [12]	28
Obr. 24 Komunikační diagram obvodu DS3231 [12]	29
Obr. 25 Adresová struktura obvodu DS3231 [12]	29
Obr. 26 Přípravek pro připojení periférií	30
Obr. 27 Schéma zapojení demonstračního přípravku	31
Obr. 28 Seznam pinů konektoru	32
Obr. 29 Vývojový diagram funkce přerušení od modulu TPM1	35
Obr. 30 Vývojový diagram funkce přerušení od modulu TPM2	36
Obr. 31 Inicializace LCD displeje	40

Seznam tabulek

Tab. 1 Výběr pinů pro signály SCL a SDA	33
Tab. 2 Závislost SCLdivider na bitech ICR	51

1. ÚVOD

Táto bakalárská práca sa zaoberá pripojením rozširujúcich periférií k vývojovej deske TWR-S08LH64, ktorá obsahuje mikrokontrolér rady MC9S08LH a návrhom dvoch laboratorných úloh demonštrujúcich použitie daných periférií. Ako rozširujúce periférie boli zvolené nasledujúce zariadenia: rotačný enkodér, maticová klávesnica, LCD displej a kalendárový obvod. Pre pripojenie rotačného enkodéru je potreba mať základné znalosti princípov manipulácie so Vstupmi/Výstupmi mikrokontroléru a ich správnej konfigurácie a použitia vhodného typu prerušenia potrebného pre optimálnu obsluhu zariadenia. S pripojením maticovej klávesnice pribude složitější logika spracovania signálov pre správnu funkčnosť zariadenia. Pripojenie LCD displeja bude realizované cez IIC prevodník s využitím IIC modulu mikrokontroléru, komunikujúci podľa štandardu sériovej sbornej IIC. Komunikácia s kalendárovým obvodom bude rovnako prebiehať podľa sbornej IIC za pomoci modulu IIC mikrokontroléru. Pre názornú ukážku použitia periférií dojde k ich pripojeniu naraz. Za pomoci enkodéru a klávesnice bude možné ovládať kalendárový obvod a za využitia LCD displeja zobrazovať všetky k tomu potrebné informácie.

Na záver budú navrhnuté dve laboratorné úlohy demonštrujúce pripojenie enkodéru a LCD displeja pre seznámenie študentov s problematikou pripojenia externých zariadení k vývojovej deske TWR-S08LH64.

2. MIKROKONTROLÉR MC9S08LH64 A VÝVOJOVÁ DESKA TWR-S08LH64

2.1 Mikrokontroler MC9S08LH64

MC9S08LH64 je osmi bitový mikrokontrolér firmy Freescale s 64 KB flash pamětí, 4 KB RAM pamětí. Podporuje 32 zdrojů přerušení. Za pomoci několika úsporných režimů se dosazuje velmi nízká spotřeba energie. Mechanismy ochrany systému mikrokontroléru:

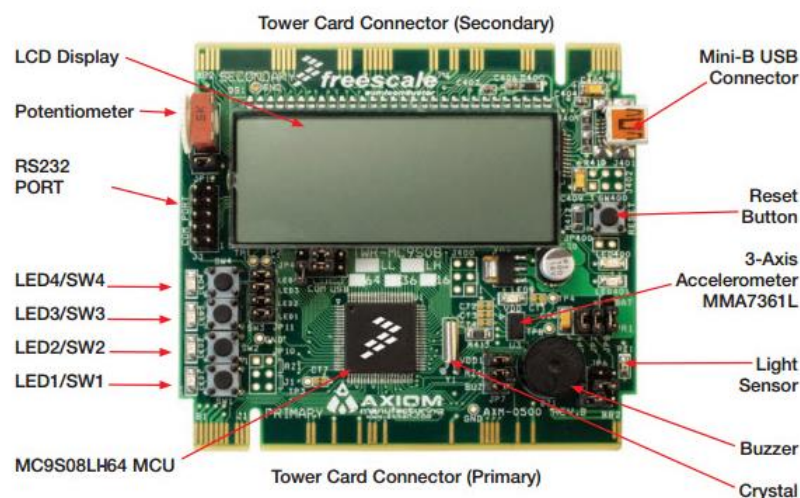
- Mechanismus resetování systému při zacyklení – Watchdog.
- Signalizace nízké úrovně napětí pomocí přerušení.
- Detekce nízké úrovně napětí s vygenerováním přerušení nebo provedením resetu systému.
- Detekce nedovoleného operačního znaku nebo adresy s provedením resetu systému.
- Ochrana oproti neoprávněnému přístupu k FLASH paměti.

MCU dokáže pracovat s kmitočtem do 40MHz při napájecím napětí 2.1 V až 3.6 V, při napětí 1.8 V až 2.1 V dokáže pracovat s kmitočtem do 20 MHz. Periferií mikrokontroléru:

- LCD driver. Pro ovládní LCD displejem.
- Osmi kanálový AD převodník ADC. Maximální rozlišení 16 bitů.
- 2 časovače TPM (Time Pulse-Width Modulator).
- TOD modul (Time of Day).
- Moduly sériových sběrnic IIC, SPI, SCI .
- Analogový komparátor ACMP. [3]

2.2 Vývojová deska TWR-S08LH64

Vývojová deska TWR-S08LH64 (Obr.1) určena pro použití s mikrokontroléry z rodiny MC9S08LH od firmy Freescale.



Obr. 1 Vývojová deska TWR-S08LH64 [1]

Deska je navržena jako modul systému Freescale Tower System (Obr.2). Freescale Tower System je modulární vývojová platforma pro 8-/ , 16-/ a 32-bitové mikrokontroléry. V závislosti na účelu vývoje lze sestavit vlastní konfiguraci systému prostřednictvím různých modulů systému.



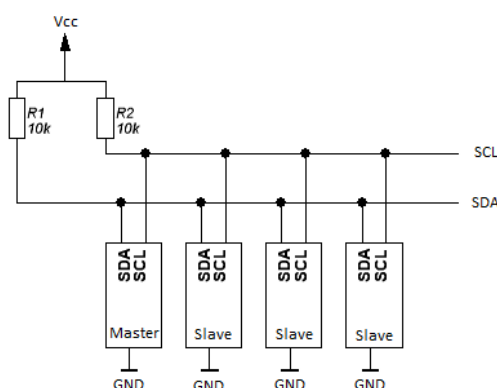
Obr. 2 Freescale Tower Systém [2]

Vývojová deska se programuje pomocí OSBDM interfejsu, který je plně podporován vývojovým prostředím CodeWarrior. Desku TRW-S08LH64 lze používat bez nadstavby Tower, avšak v tomto případě nemá programátor přístup ke všem portům mikrokontroleru. Pro studijní účely v rámci seznámení studentů se základy programování mikrokontrolérů v plné míře postačí samostatná deska s již existujícími periferiemi, tedy těmi, jež se nachází na desce:

- LCD display GD-5306P je přes 30 pinů je přímo připojen k mikrokontroléru.
- tříosý akcelerometr MMA7361L.
- piezo-bzučák s rezonančním kmitočtem 2400 Hz.
- senzor světla, který je realizován jako fototranzistor. Výstupní proud fototranzistoru je závislý na intenzitě světla, které dopadá na povrch.
- 4 uživatelská tlačítka, kdy ke každému tlačítku je připojen 22 pF kondenzátor pro ošetření zámků.
- 4 uživatelské LED diody. Diody jsou zelené barvy a jsou aktivní v logické nule. Pro omezení protékajícího proudu pro každou diodu v sérii zapojen rezistor.
- 5 k Ω potenciometr, jež slouží pro simulaci analogového vstupu. [3]

3. IIC SBĚRNICE

IIC sběrnice byla navržena firmou Philips. Pro přenos dat se používá jen dvou signálových vodičů, tj. SCL vodič, který přenáší hodinový signál a SDA vodič, který přenáší data. SCL signál generuje Master zařízení, Slave zařízení jen reaguje. SDA vodič přenáší data v obou směrech. Vytáhnutí vodiče na logickou 1 se provádí pomocí připojeného pull-up rezistoru. Optimální hodnota pull-up rezistoru je 10 k Ω . Stáhnutí se provádí připojením na zem. [4]



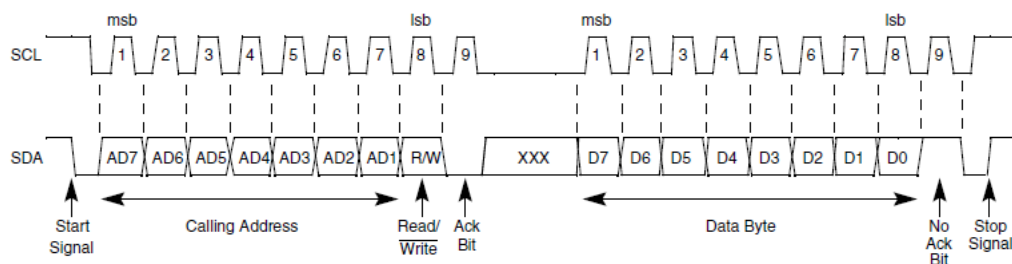
Obr. 3 Příklad zapojení zařízení při použití IIC sběrnice [4]

Standartní komunikace se skládá z 4 částí:

- Start signál
- Odesílání adresy Slave zařízení
- Přenos dat
- Stop signál [3]

3.1.1 Start signál

Když je sběrnice svobodná (SCL a SDA vodiče jsou v logické 1), Master zařízení může začít komunikaci pomocí vygenerování Start signálu. Start signál je definován stáhnutím vodiče SDA z logické 1 do logické 0 při logické 1 signálu SCL. Po vygenerování Start signálu, lze začít přenos dat. [3]



Obr. 4 Komunikace po IIC sběrnici [3]

3.1.2 Odesílání adresy Slave zařízení

Počínaje prvním bajtem přenosu dat až po vygenerování Start signálu musí být stanoveno, se kterým Slave zařízením, a ve kterém směru bude probíhat následující komunikace. V závislosti na způsobu adresace se utváří postup. Obecně IIC sběrnice definuje 7-bitovou a 10-bitovou adresaci. Při 7-bitové adresaci, obsahuje první bajt posílaný po Start signálu RW bit, který určí směr přenosu dat (1- Read transfer, Master bude přijímat data od Slave zařízení. 0 – Write transfer, Master bude odesílat data Slave zařízení) a 7 bitů adresy Slave zařízení. Ze sedmi bitů lze vygenerovat 127 unikátních adres Slave zařízení, z toho plyne, že do IIC sběrnice při 7-bitové adresaci lze zapojit jen 127 Slave zařízení.

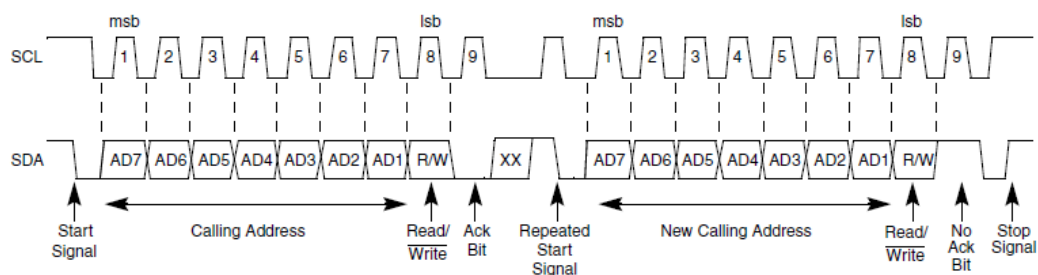
Při 10-bitové adresaci určení Slave zařízení a směru komunikace probíhá pomocí odeslání dvou bajtů. První bajt obsahuje RW bit, dva MSB bity adresy Slave zařízení a tzv. identifikátor 10-bitové adresace. Druhý bajt obsahuje zbývajících 8 bit adresy Slave zařízení. Do IIC sběrnice lze zapojit 1023 zařízení, které komunikují podle 10-bitové adresace. Po odeslání Masterem adresy Slave zařízení, Slave zařízení s odpovídající adresou pošle tzv. *Acknowledge* bit, který v tomto případě signalizuje úspěšné přijetí 1 bajtu dat. *Acknowledge* bit se definuje stáhnutím SDA na logickou 0 při devátém taktu hodinového impulsu na vodiči SCL. [3]

3.1.3 Přenos dat

Po úspěšném spojení Master a Slave zařízení se sleduje přenos dat. V závislosti na tom, který bit byl poslán při odesílání adresy Slave zařízení ($R=1$ nebo $W=0$) určuje se směr přenosu dat. Všechna data jsou 8-bitová. MSB bit se posílá první. Při komunikaci Master Out Slave In (Master odesílá data Slave zařízení), po přijetí dat Slave zase vygeneruje *Acknowledge* bit. Jestli Master neobdrží *Acknowledge* bit od Slave zařízení, bude odesílání považováno za neúspěšné. Při komunikaci Master In Slave Out (Master přijímá data od Slave zařízení): jestli Master přijme 1 bajt dat od Slave zařízení a bude chtít i dále číst data, Master musí vygenerovat *Acknowledge* bit, protože Slave zařízení zhodnotí nepřijetí *Acknowledge* bitu jako konec přenosu dat a uvolní vodič SDA. [3]

3.1.4 Stop signál

Master může zrušit komunikaci pomocí vygenerování Stop signálu pro uvolnění sběrnice. Stop signál je definován stáhnutím vodiče SDA při nízké úrovni SCL vodiče. Master může vygenerovat Stop signál jen v případě, že Slave zařízení odeslalo *Acknowledge* bit, který signalizuje ukončení aktivity ze strany Slave zařízení. V průběhu komunikace, tedy před generováním Stop signálu, Master zařízení může vygenerovat Start signál, tento signál se nazývá Repeated Start signál. Master generuje takový signál v případě, že chce komunikovat s jiným Slave zařízením nebo se stejným zařízením ale v jiném směru. [3]



Obr. 5 Komunikace po IIC sběrnici s Repeated Start signálem [3]

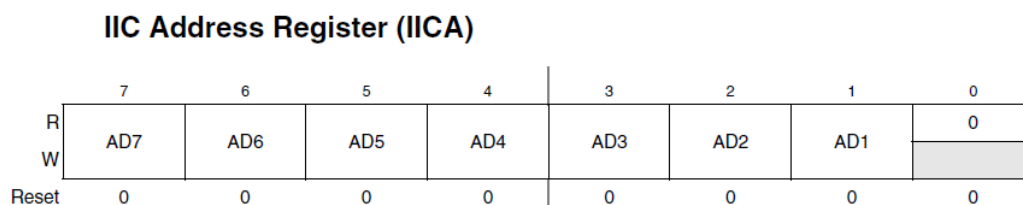
3.2 Modul IIC mikrokontroléru MC9S08LH64

V této kapitole se provede rozbor modulu IIC mikrokontroléru MC9S08LH64RM, popíšou se registry, jejich význam.

Modul IIC mikrokontroléru MC9S08LH64 obsahuje 6 řídicích a stavových registrů. Maximální rychlost sběrnice při plné zátěži je 100 kbps. Počet možných připojených zařízení a délka sběrnice jsou omezené kapacitou 400 pF. Adresa mikrokontroléru je programovatelná. [3]

3.2.1 IICA (IIC Address Register)

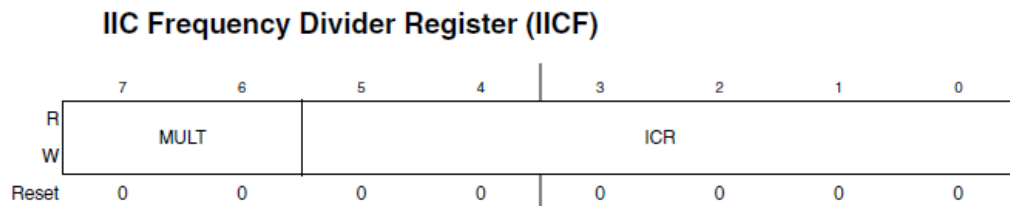
Registr IICA slouží pro chránění adresy mikrokontroléru, když pracuje v režimu Slave. [3]



Obr. 6 IICA registr [3]

3.2.2 IICF (IIC Frequency Divider Register)

Pomocí registru IICF lze nastavit Baud Rate IIC sběrnice. Baud Rate je to veličina, která udává počet možných změn stavu signálu za jednotku času. Jinými slovy udává rychlost přenosu dat. [3]



Obr. 7 IICF registr [3]

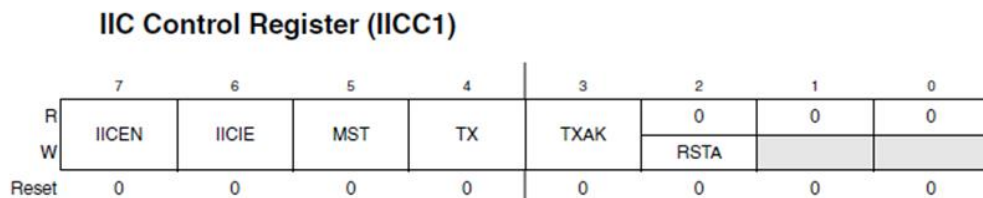
MULT bity v kombinaci s ICR bity nastavují IIC baud rate. Hodnotu IIC baud rate lze vypočítat podle vztahu:

$$\text{IIC baud rate} = \frac{\text{bus speed (Hz)}}{2 * \text{mul} * \text{SCLdivider}};$$

Parametr SCLdivider je definován bity ICR. Pomocí Tab.2(viz. příloha) lze stanovit jakou hodnotu bude mít parametr SCLdivider v závislosti na bitech ICR. Standartní hodnota Baud rate sběrnice IIC je 100 kbps. [3]

3.2.3 IICC1 (IIC Control Register)

V registru IICC1 se provádí nejdůležitější nastavení IIC modulu.



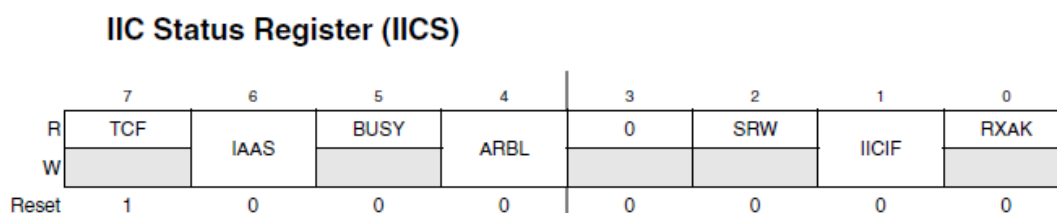
Obr. 8 IICC1 registr [3]

Bit IICEN zapíná/vypíná modul IIC. IICEN = 0 modul vypnut, IICEN = 1 modul zapnut. Bit IICIE povolí nebo zakáže přerušení od modulu IIC. IICIE = 0 přerušení zakázáno, IICIE = 1 přerušení povoleno. MST bit určuje ve kterém režimu bude pracovat mikrokontrolér. Jestli MST = 0 mikrokontrolér bude pracovat ve režimu Slave, MST = 1 mikrokontrolér pracuje v režimu Master. Bit MST taky se používá pro generování Start a Stop signálu. Když bit MST mění hodnotu z 0 na 1, generuje se Start signál. Naopak když MST mění hodnotu z 1 na 0, generuje se Stop signál. Bit TX určuje směr komunikace mezi Master a Slave zařízením. TX = 0, Master přijímá data od Slave zařízení. TX = 1 Master odesílá data Slave zařízení. Bit TXAK stanovuje, jestli bude generován *Acknowledge* bit v režimu přijetí dat. Když TXAK = 0 *Acknowledge* bit

bude odeslán po přijetí 1 bajtu dat a přenášení dat se pokračuje, když TXAK = 1 Acknowledge bit nebude generován a přijetí dat se ukončí. Zapsáním 1 do RSTA bitu se generuje Repeated Start signál. [3]

3.2.4 IICS (IIC Status Register)

Tento registr slouží pro indikaci stavu jednotlivých událostí.



Obr. 9 IICS registr [3]

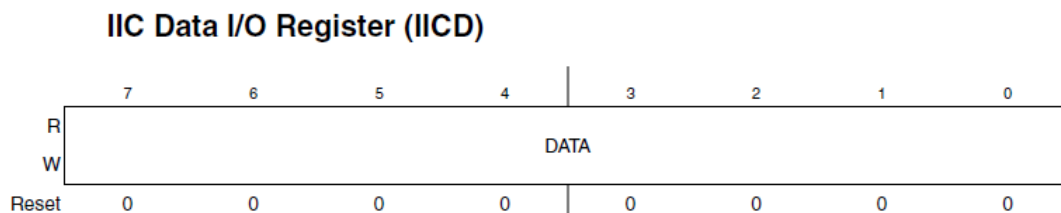
TCF bit nastaven po dokončení přenosu 1 bajtu dat. Bit je platný pouze během nebo ihned po přenosu. Bit se vymaže čtením IICD registru při přijetí dat, nebo zápisem do IICD při odesílání dat. IAAS bit se nastaví, když mikrokontrolér jako Slave zařízení uslyší svou adresu nebo jestli General Call Address je aktivní a General Call Address je přijat. General Call Address je proces adresace Masterem všech Slave zařízení, které jsou připojené ke sběrnici, pomocí adresy 0x00. Slave zařízení, které umí přijímat data, odpoví *Acknowledge* bitem a dále pracují v režimu Slave Receiver. Dále jde odesílání dat od Master zařízení ke Slave zařízení, které musí přečíst data a odeslat *Acknowledge* bit, pokud to umí. Význam General Call Address se spočívá v datech, jdoucích po General Call Address. Nejčastěji jsou data komandy ke resetu, očištění paměti atd. Zapsáním do registru IICC bit IAAS se očisti. BUSY bit indikuje stav sběrnice. Bit se nastavuje po odeslání nebo přijetí Start signálu a očisti se po odeslání nebo přijetí stop signálu. ARBL bit se nastavuje, když Arbitration procedura je ztracena. Data Arbitration procedure je proces určení priority mezi Master zařízeními při zapojení do IIC sběrnice několika Master zařízení (Multi-master bus). Bit se očisti zapsáním do něho 1. SRW bit ve Slave modu indikuje hodnotu R/W bitu přijatého se svou adresou. Jestli SRW má hodnotu 0 Slave zařízení bude přijímat data, Master odesílat. Jestli SRW je roven 1, Master přijímá data, Slave odesílá. Dalším bitem v registru IICS je IICIF bit, který se nastaví, když se očekává přerušení. Jinými slovy tento bit se nastavuje, když jsou splněné podmínky pro vyvolání přerušení. Podmínkami ke vyvolání přerušení jsou:

- Přenášení 1 bajtu dat
- Shoda přijaté adresy v Slave režimu
- Arbitration lost

Tedy se musí říct, že bit se nastavuje nezávisle na tom, jestli přerušení povoleno, nebo ne. Pro očištění IICIF bitu je potřeba zapsat do něho 1. RXAK bit indikuje přijetí *Acknowledge* bitu. Jestli *Acknowledge* bit přijat bit RXAK má hodnotu 0, v opačném případě 1. [3]

3.2.5 IICD (IIC Data I/O Register)

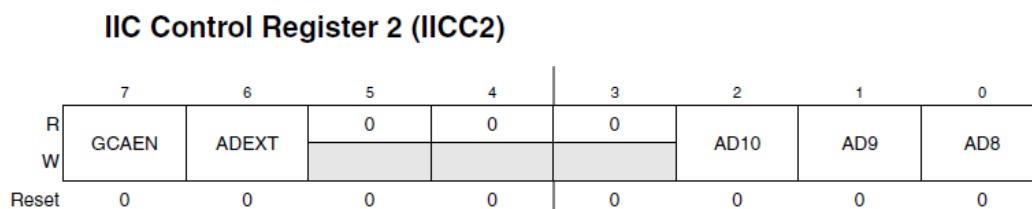
Registr slouží pro přenos dat. Když mikrokontroler je nastaven jako Master a směr komunikace je odesílání dat, zapsáním dat do tohoto registru se zahájí odesílání 1 bajtu dat ke Slave zařízení. Jestli směrem komunikace je přijetí dat, čtení dat z registru IICD zahájí přijetí 1 bajtu dat od Slave zařízení. MSB bit se posílá první. [3]



Obr. 10 IICD registr [3]

3.2.6 IICC2 (IIC Control Register 2)

Registr IICC2 obsahuje rozšířené nastavení modulu IIC. Pomocí bitu GCAEN lze povolit, nebo zablokovat General Call Address. ADEXT bitem lze vybrat způsob adresace. Daný mikrokontrolér podporuje 7-bitovou a 10-bitovou adresaci. Bity AD8 až AD9 obsahují horní 3 bity adresy Slave zařízení, když je vybraná 10-bitová adresace. [3]



Obr. 11 IICC2 registr [3]

4. EXTERNÍ PERIFERIE POUŽITÉ PRO REALIZACI LABORATORNÍCH ÚLOH

V této kapitole bude proveden teoretický rozbor 4 zařízení, které budou připojené k mikrokontroléru. Rozbor bude proveden na základě dostupné dokumentace. Pro realizaci zapojení externích periférií do mikrokontroléru byly vybrány následující zařízení:

- Inkrementální rotační enkodér
- Maticová klávesnice 4x4
- LCD displej 20x04 s IIC převodníkem
- Kalendářový obvod DS3231

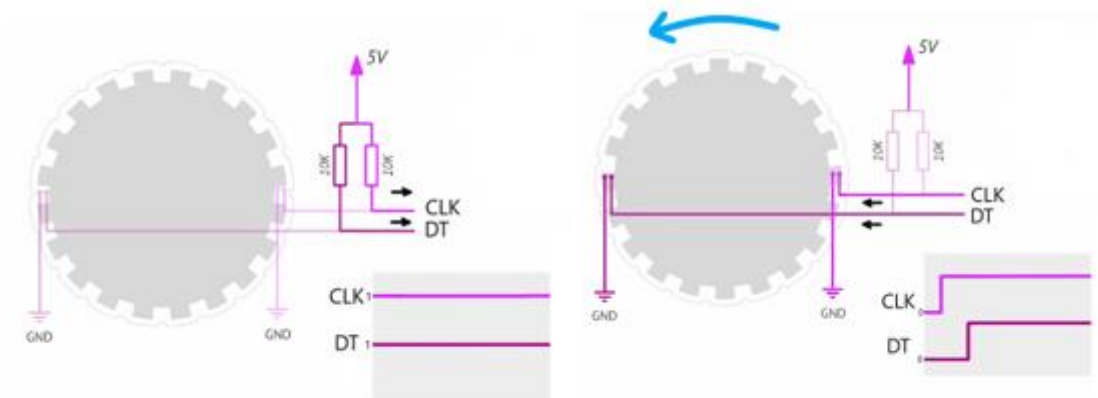
4.1.1 Inkrementální rotační enkodér

Inkrementální rotační enkodér je to zařízení, které převádí rotační pohyb hřídele do sérií elektrických impulzů (Obr. 12). Generované impulzní signály jsou 2 (CLK a DT). Signály jsou posunuté vůči sobě a díky tomu lze stanovit směr otáčení hřídele enkodéru.



Obr. 12 Enkodér [7]

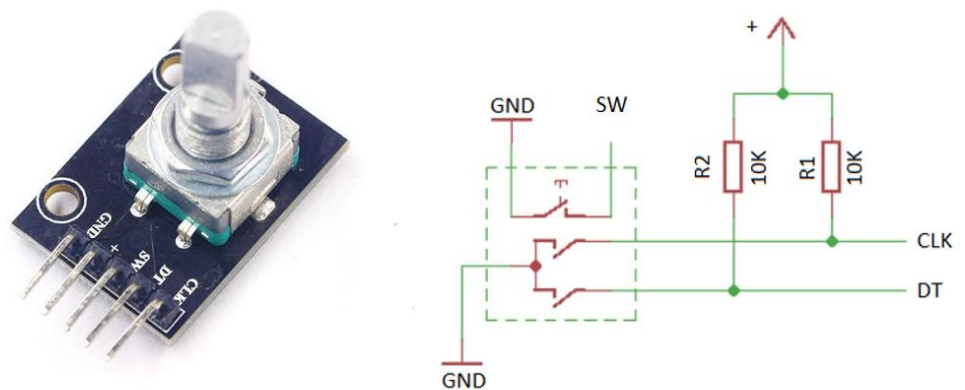
Základem inkrementálního enkodéru jsou pár kontaktů a ozubené kovové kolečko, které je spojené z hřídele. Při otáčení hřídele kontakty postupně se spojí a rozpojí, tím se generují impulzní signály. Posunutí výstupních signálů vůči sobě je realizované rozmístěním kontaktů. Obecně lze vygenerování signálů představit podle následujícího Obr. 13. [7]



Obr. 13 Průběh signálů enkodéru [7]

Pro realizaci laboratorních úloh byl vybrán modul KY-040 který obsahuje enkodér s tlačítkem a 2 pull-up rezistory. Schéma modulu je na Obr. 14. Modul s enkodérem obsahuje 5 vývodů:

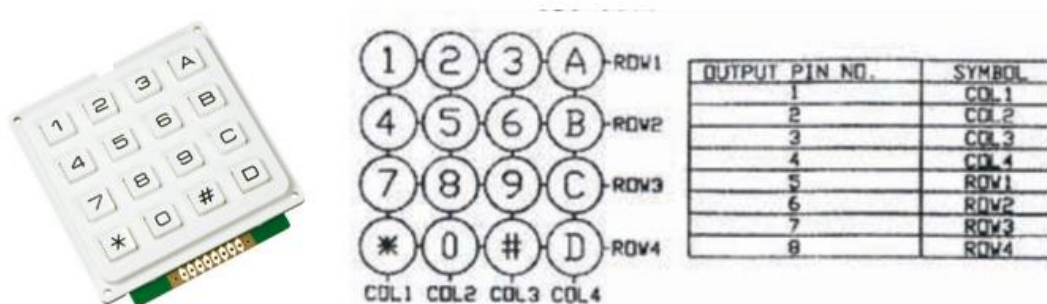
- CLK, první výstupní impulzní signál
- DT, druhý výstupní impulzní signál
- SW, vývod tlačítka enkodéru
- +, napájení enkodéru (+)
- GND, napájení enkodéru (-) [7]



Obr. 14 Modul KY-040

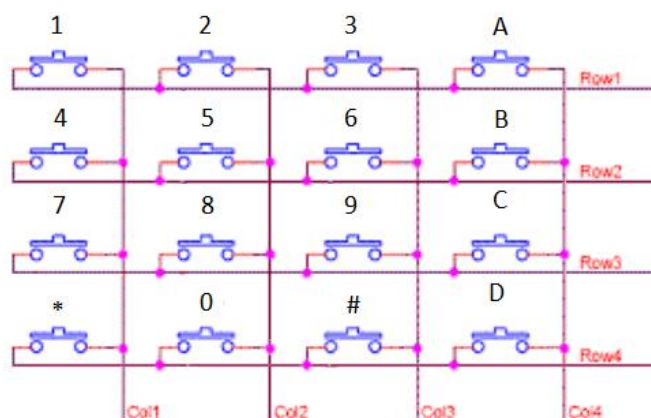
4.1.2 Maticová klávesnice F-KV16KEY

Pro realizaci přípravku byla vybrána maticová klávesnice F-KV16KEY s 16 tlačítky. Klávesnici lze napájet maximálním napětím 24 V, maximální proud je 20 mA a maximální odpor sepnutých kontaktů je 200 Ω . [8]



Obr. 15 Maticová klávesnice F-KV16KEY [8]

Schéma zapojení pro klávesnice je na Obr. 16.

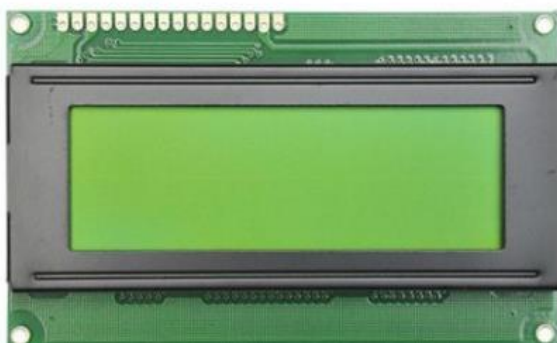


Obr. 16 Maticové zapojení klávesnice [9]

Princip snímání spočívá v tom, že 4 port piny mikrokontroléru, které jsou připojené ke sloupcům klávesnice jsou udržované na vysoké úrovni pomocí připojení pull-up rezistorů. Radky postupně se uzemní a při stisknutí tlačítka příslušný sloupec se uzemní a na vstupu mikrokontroléru se objeví nízká úroveň. Podle informace o tom, který řádek byl uzemněn a stavu signálů na vstupech mikrokontroléru lze zjistit, které tlačítko bylo stisknuto. Popsané zapojení funguje při současném stisknutí dvou tlačítek. V případě současného stisknutí třech a více tlačítek se dochází k chybné detekci. Hlavní výhodou maticového zapojení je to, že pomocí 8 Vstupů/Výstupů mikrokontroléru lze realizovat obsluhu 16 tlačítek. Při zmáčknutí tlačítek klávesnice dochází k zákmitům, které je třeba softwarově ošetřit. [9]

4.1.3 LCD displej 20x04 s IIC převodníkem

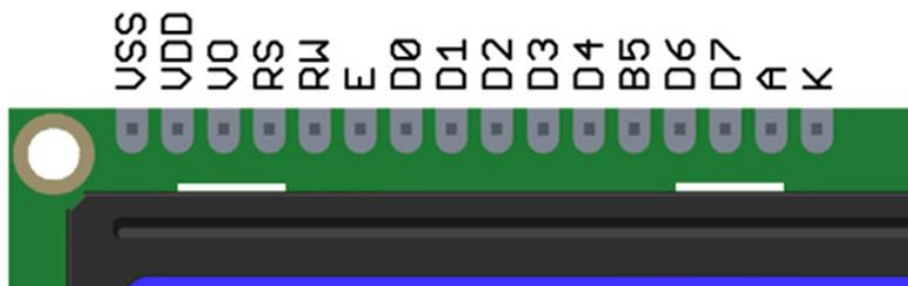
Žlutý LCD displej s podsvícením na bázi řadiče HD44780. Obsahuje 4 řádky s délkou 20 znaků. Rozlišení znaku 5x8 pixelů při použití 4 řádků, 5x10 pixelů při použití 2 řádků. Jeden z nejjednodušších a nejdostupnějších displejů pro realizaci projektů s využitím mikrokontroléru.



Obr. 17 LCD displej 2004a [10]

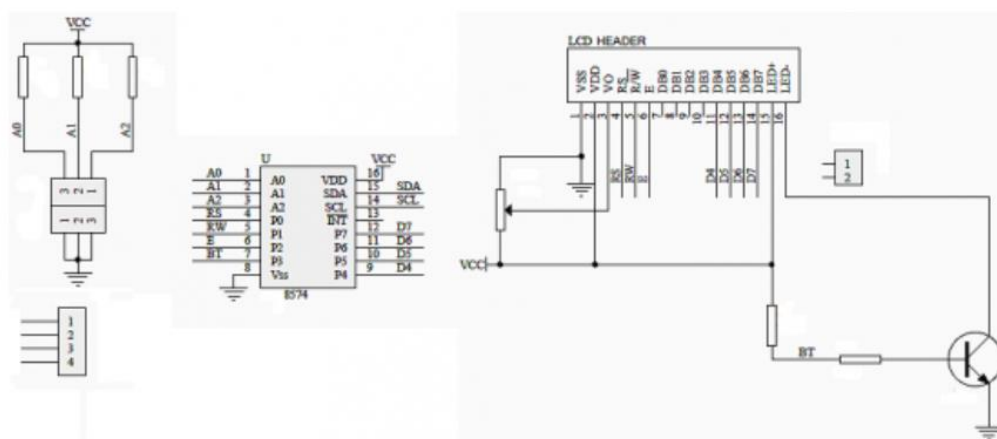
Interface displeje je tvořen následujícími 16 vývody:

- 1 VSS – napájení displeje (-)
- 2 VDD – napájení displeje (+), napětí 5 V
- 3 VO – vývod pro regulaci kontrastu displeje, vývod je potřeba zapojit na potenciometr 10 k Ω , výstupním napětí z potenciometru se da regulovat úroveň kontrastu displeje.
- 4 RS – vyber registru (registr dat nebo registr instrukci)
- 5 RW – čtení/zápis
- 6 E – enable pin
- 7-10 D0-D3 - dolní půlbajt, používá se jen pro 8-bitovy interfejs
- 11-14 D4-D7 – horní půlbajt
- 15 A – anoda napájení LED podsvícení
- 16 K – katoda napájení LED podsvícení [5]



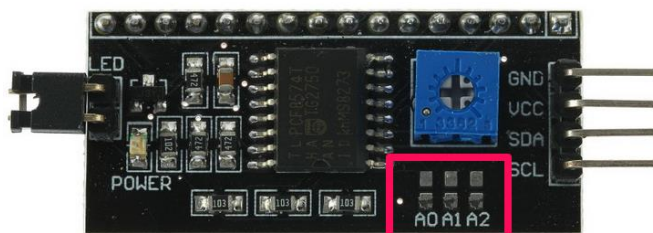
Obr. 18 Umístění vývodů LCD displeje

Pro ovládní displejem je potřeba dodržovat správné časování signálů. V dokumentaci pro displeji lze najít časový diagram, který definuje správné pořadí a hodnoty zpoždění signálů vůči sobě. Displej umožňuje přenášet data podle 8- a 4-bitového interfejsu. Při 8-bitovém interfejsu se přenos 1 bajtu dat realizuje pomocí všech 8 datových vývodů (D0-D7). 4-bitový interfejs pro přenos využívá jen 4 datových vývodů (D4-D7) a přenos 1 bajtu je rozdělen na přenos dvou půlbajtů. 8-bitový interfejs je rychlejší pro přenášení dat, ale kvůli omezení aktualizace pixelů displeje je tato výhoda je zanedbatelná z hlediska zobrazení, ale může být užitečná pro software. V dané bakalářské práci LCD displej k mikrokontroléru připojen pomocí IIC sběrnice. Pro realizace byl využit převodník display – IIC. Tento převodník využívá integrovaný obvod PCF8574 a lze jej zakoupit např. v GM electronic. Schéma převodníku a jeho připojení k displeji je na Obr. 19.



Obr. 19 Připojení IIC převodníku k LCD displeji [6]

Pin P0 je připojen na vývod RS LCD displeje, pin P1 je připojen na vývod RW, pin 2 na vývod E, pin 3 na vývod A, piny 4-7 na vývody D4-D7. Proto po odeslání 1 bajtu dat do IIC převodníku bajt se objeví na pinech P0-P7, které jsou připojené na příslušné vývody LCD displeje. Takže 1 bajt odeslaný do IIC převodníku, objeví se na vývodech RS, RW, E, A, D4-D7 LCD displeje. Převodník obsahuje potenciometr pro nastavení kontrastu displeje, jumper, kterým je možné zapínat nebo vypínat LED podsvícení displeje, 3 piny A0, A1, A2, které se používají pro změnu adresy převodníku. Defaultní adresa převodníku je 0x3F. [6]



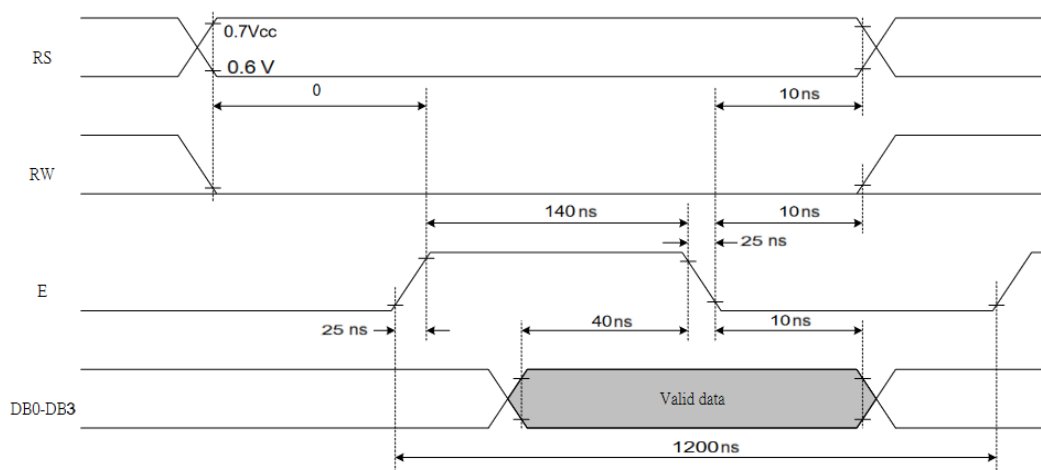
Obr. 20 Piny A0, A1, A2 IIC převodníku [6]

Pomocí 3 pinů lze sestavit 8 kombinací, kterými lze zakódovat 8 unikátních adres IIC převodníku. Interfejs IIC převodníku je tvořen 4 vývody:

- GND, napájení převodníku (-)
- VCC, napájení převodníku (+)
- SDA, vývod pro datový vodič IIC sběrnice
- SCL, vývod pro hodinový vodič IIC sběrnice [6]

Komunikace s LCD displejem probíhá podle protokolu, popsaného v dokumentaci k displeji. S využitím IIC převodníku proces odeslání 1 bajtu do LCD displeje vypadá následujícím způsobem:

- 1) Mikrokontrolér generuje Start signál pro začátek komunikace.
- 2) Mikrokontrolér odešle adresu IIC převodníku doplněnou o RW bit, který je roven nule (zápis). IIC převodník odpoví *Acknowledge* bitem, signalizující úspěšné přijetí 1 bajtu dat.
- 3) Mikrokontrolér odešle 1 bajt dat do IIC převodníku tak, aby na vývodu RS byla nastavena 0, jestli mikrokontrolér odesílá komandu do instrukčního registru, nebo 1, jestli odesílá znak v ANSI kódu do datového registru. Na vývodu RW byla nastavena 0 pro zápis dat, na vývodu E byla nastavena 0, protože v této fázi je potřeba jen nachystat data na datových vývodech LCD displeje, na vývodu A byla nastavena 1 pro zapnutí LED podsvícení, na vývodech D4-D7 byl nastaven horní půlbajt komandy nebo znaku.
Po úspěšném přijetí 1 bajtu dat, IIC převodník odpoví *Acknowledge* bitem.
- 4) Mikrokontrolér pošle stejný bajt jen s tím rozdílem, že na vývodu E se musí objevit 1, čím se povolí přijetí dat LCD displejem. Po úspěšném přijetí 1 bajtu dat, IIC převodník odešle *Acknowledge* bit. Vygeneruje se zpoždění podle časového diagramu (Obr.21).
- 5) Mikrokontrolér zase pošle stejný bajt, jen pro ukončení odeslání dat do LCD displeje na vývodu E musí se nastavit 0. Po úspěšném přijetí 1 bajtu dat, IIC převodník odešle *Acknowledge* bit. Vygeneruje se zpoždění.
- 6) Na daný moment je odeslán horní půlbajt komandy nebo znaku do LCD displeje. Body 3-5 se opakují pro odeslání dolního půlbajtu komandy nebo znaku.
- 7) Mikrokontrolér vygeneruje Stop signál pro ukončení komunikaci.



Obr. 21 Časový diagram LCD displeje [5]

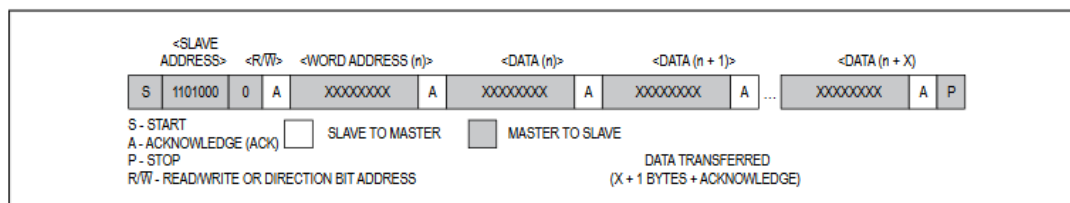
4.1.4 Modul ZS-042 s kalendářovým obvodem D3231

Poslední zařízení, které bylo připojeno k mikrokontroléru je kalendářový obvod.

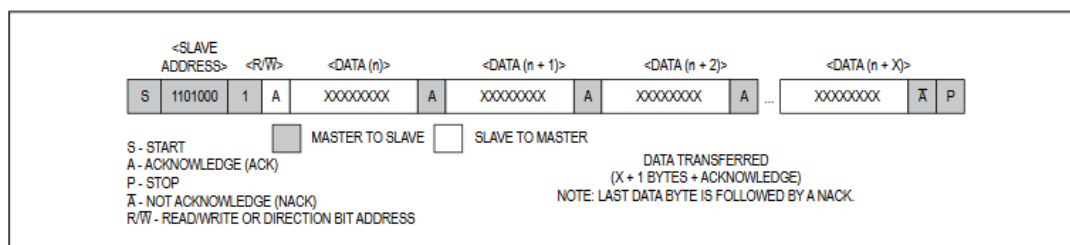


Obr. 22 Modul ZS-042 [11]

Kalendářový obvod je zařízení, které udržuje informace o hodinách, minutách, sekundách, dnech, dnů, roku a dnů v týdnu. Obvod automaticky přizpůsoben pro měsíce, ve kterých miň než 31 den a obsahuje korekce pro přestupný rok. Hodinový formát lze přepínat mezi 12 a 24. Pro 12 hodinový formát lze uložit příznak AM/PM. Pro připojení byl vybrán modul ZS-042, který obsahuje kalendářový obvod DS3231, paměť EEPROM 32K a veškeré potřebné pomocné součástky včetně záložního napájení, které realizované pomocí knoflíkové baterky CR2032. Stejně jako IIC převodník, modul ZS-042 obsahuje piny A0, A1, A2 pro změnu adresy zařízení. Defaultní adresa modulu je 0x68. Schéma zapojení je znázorněné na Obr.23.



Data Write—Slave Receiver Mode



Data Read—Slave Transmitter Mode

Obr. 24 Komunikační diagram obvodu DS3231[12]

Data ve většině případu jsou uloženy v BCD formátu. Paměťová struktura obvodu je uvedena na Obr. 25.

ADDRESS	BIT 7 MSB	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0 LSB	FUNCTION	RANGE
00h	0	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12/24	AM/PM 20 Hour	10 Hour	Hour				Hours	1–12 + AM/PM 00–23
03h	0	0	0	0	0	Day			Day	1–7
04h	0	0	10 Date		Date				Date	01–31
05h	Century	0	0	10 Month	Month				Month/ Century	01–12 + Century
06h	10 Year				Year				Year	00–99
07h	A1M1	10 Seconds			Seconds				Alarm 1 Seconds	00–59
08h	A1M2	10 Minutes			Minutes				Alarm 1 Minutes	00–59
09h	A1M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 1 Hours	1–12 + AM/PM 00–23
0Ah	A1M4	DY/DT	10 Date		Day				Alarm 1 Day	1–7
			Date		Date				Alarm 1 Date	1–31
0Bh	A2M2	10 Minutes			Minutes				Alarm 2 Minutes	00–59
0Ch	A2M3	12/24	AM/PM 20 Hour	10 Hour	Hour				Alarm 2 Hours	1–12 + AM/PM 00–23
0Dh	A2M4	DY/DT	10 Date		Day				Alarm 2 Day	1–7
			Date		Date				Alarm 2 Date	1–31
0Eh	EOSC	BBSQW			CONV	RS2	RS1	INTCN	A2IE	A1IE
0Fh	OSF	0	0	0	EN32kHz	BSY	A2F	A1F	Control/Status	—
10h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	Aging Offset	—
11h	SIGN	DATA	DATA	DATA	DATA	DATA	DATA	DATA	MSB of Temp	—
12h	DATA	DATA	0	0	0	0	0	0	LSB of Temp	—

Obr. 25 Adresová struktura obvodu DS3231 [12]

5. PŘÍPRAVEK PRO PŘIPOJENÍ PERIFERIÍ

Kvůli tomu, že periférií vyplňují funkce, které vzájemně se doplňují (Kalendářový obvod poskytuje data, LCD displej zobrazuje data, enkodér a klávesnice řídí a nastavují data), pro demonstrace připojení bylo řešeno připojit všechny zařízení najednou. Dále kvůli tomu, že některé periférií obsahují vývody se stejnými funkce, bylo řešeno vytvořit přípravek, který by obsahoval vše zařízení a vývody, které mají stejné funkce, by byli spojené mezi sebou a vyvedené na konektor. Daný přípravek byl navržen a znázorněn na Obr.26



Obr. 26 Přípravek pro připojení periférií

Kalendářový obvod poskytuje informace o čase a datumu. Pomocí enkodéru a klávesnici se realizuje nastavení hodnot času a daty kalendářového obvodu. Pomocí LCD displeje se zobrazuje aktuální čas, datum a veškerá informace potřebná pro nastavení dat do obvodu. Přípravek obsahuje pomocné součástky: LED dioda s

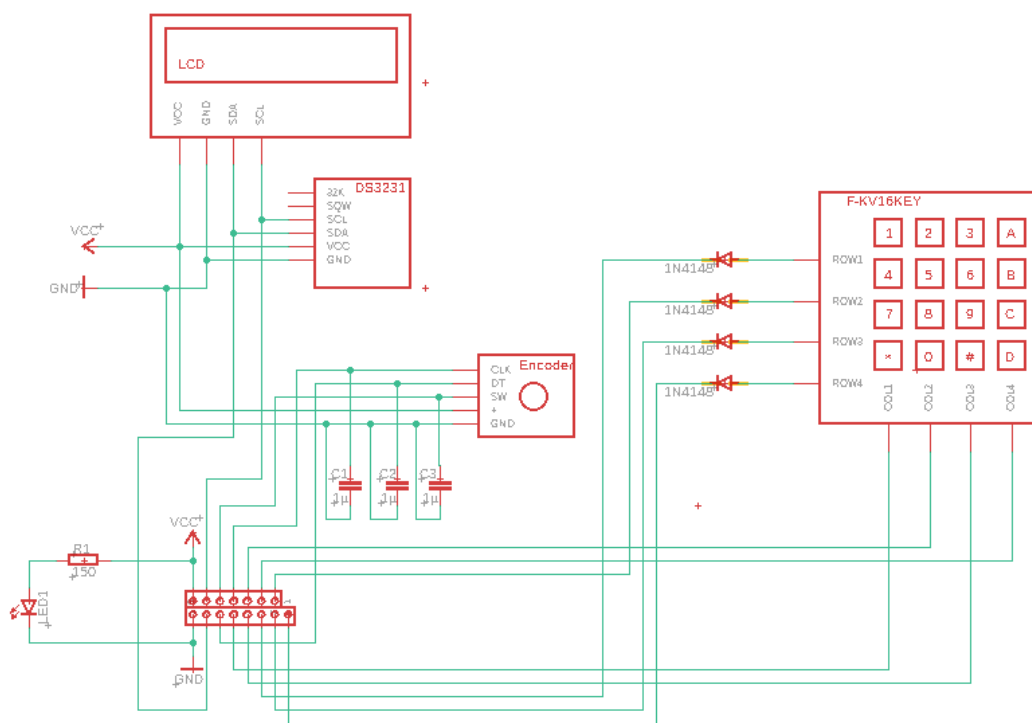
rezistorem $150\ \Omega$ indikující přítomnost napětí, diody 1N4148 pro zajištění správného směru proudu, kondenzátory $1\ \mu\text{F}$ pro ošetření zákmitů kontaktů enkodéru. Kapacita kondenzátoru jsem zvolil podle požadované časové konstanty náběžné/sestupné hrany. Z dokumentaci k desce TWR-S08LH64 lze zjistit hodnotu přepisovatelného pull-up rezistoru. Hodnota daného pull-up rezistoru je $10\ \text{k}\Omega$. Vztah pro výpočet časové konstanty RC článku je:

$$\tau = RC;$$

Proto vypočet hodnoty kapacity kondenzátoru při požadované hodnotě časové konstanty $1\ \text{ms}$:

$$C = \frac{\tau}{R} = \frac{1 * 10^{-3}}{10 * 10^3} = 0,1\ \mu\text{F};$$

Schéma zapojení je znázorněna na Obr. 27



Obr. 27 Schéma zapojení demonstračního přípravku

V levé dolní části přípravku se nachází konektor, určený pro připojení k mikrokontroléru. Význam každého pinu konektoru je znázorněn na Obr. 28



Obr. 28 Seznam pinů konektoru

5.1.1 Připojení přípravku k desce

V první části této kapitoly jsou zmíněné obecné požadavky na připojení k mikrokontroléru. V druhé části je popsáno, jakým způsobem jsem ho zapojil do desky TWR-S08LH64.

Na přípravku celkem 15 pinů. Pin VCC je pin pro napájení a musí být připojen na zdroj napětí 5 V. Pin GND musí být připojen na zem. Pin SCL je určen pro hodinový vodič sběrnice IIC. Pin SDA je pin pro datový vodič sběrnice IIC. Pin SW je pin pro tlačítko enkodéru, který je potřeba zapojit na vstupní pin mikrokontroléru, který podporuje modul KBI, s připojením pull-up rezistoru. Pin CLK je jeden z výstupních impulzních signálů enkodéru, je potřeba připojit ho na vstupní pin mikrokontroléru. Pull-up rezistor už je navržen na přípravku, proto není potřeba ho tam připojovat (Obr. 14). Pin DT je druhý výstupní impulzní signál enkodéru, posunutý o 90° vůči signálu CLK. Je potřeba zapojit ho na vstupní pin mikrokontroléru. Pull-up rezistor už je navržen na přípravku. Piny COL1-COL4 je nutné zapojit na vstupní piny mikrokonroléru s podporou modulu KBI a připojením pull-up rezistorů. Piny ROW1-ROW4 budou postupně se uzemňovat a je potřeba jich zapojit na výstupní piny mikrokontroléru.

Do desky TWR-S08LH64 přípravek lze připojit pouze s využitím nadstavby TOWER z toho důvodu, že na samotné desce není dostup k potřebným portům mikrokontroléru. Nadstavba TOWER se skládá ze dvou karet: primární a sekundární. Primární karta se připojuje ke primárnímu konektoru desky, sekundární ke sekundárnímu konektoru. Na každé kartě jsou 2 konektory pro přístup k signálům a portům mikrokontroléru. Na primární kartě konektory J5 a J6. Na Sekundární kartě konektory J8 a J9. V dokumentaci k desce TWR-S08LH64 (schémata) jsou znázorněné dva sloty: J1 – primární, J2 – sekundární (Obr. 29). Levá strana slotu J1 odpovídá konektoru J8, pravá strana odpovídá konektoru J9. Levá strana slotu J2 odpovídá konektoru J8, pravá strana konektoru J9. Připojení k požadovanému pinu probíhá následujícím způsobem:

- 1) Ve schématech k desce TWR-S08LH64 je potřeba zjistit číslo pinu, slot a stranu slotu na kterém se nachází požadovaný vývod mikrokontroléru.
- 2) Na základě slotu a strany slotu se zjistí karta a konektor na kartě. Na základě čísla pinu se zjistí pin, na kterém dany vývod je dostupný.

Pin VCC podle schémat k desce jsem zapojil na pin B1 konektoru J9, který odpovídá signálu +5V_ELEV. Pin GND jsem zapojil na pin B2 konektoru J9, který odpovídá signálu GND. Pin SCL jsem zapojil na pin B10 konektoru J9, který odpovídá port pinu PTB5. Piny pro signály SCL a SDA lze zvolit pomocí bitu IICPS v registru SOPT2 (Tab. 2). Defaultně pinu SCL odpovídá port pin PTB5 a pinu SDA port pin PTB4. Pin SDA jsem zapojil na pin B11 konektoru J9, který odpovídá port pinu PTB4.

IICPS in SOPT2	Port Pin for SDA	Port Pin for SCL
0 (default)	PTB4	PTB5
1	PTA2	PTA3

Tab. 1 Výběr pinů pro signály SCL a SDA

Pin SW jsem zapojil na pin A27 konektoru J8, který odpovídá port pinu PTA6. Port pin PTA6 jsem nakonfiguroval jako vstupní s připojením pull-up rezistoru. Port A byl vybrán z toho důvodu, že podporuje modul KBI. Pin DT jsem připojil na pin A42 konektoru J8, který odpovídá port pinu PTC1. Pin CLK jsem připojil na pin 2 konektoru JP12, který odpovídá port pinu PTA0. Dane port piny jsem nakonfiguroval jako vstupní bez připojení pull-up rezistoru, protože na přípravku KY-040 pull-up rezistory už navržené. Piny COL1-COL4 jsem zapojil na piny B58, B56, B55 a B57 konektoru J9, které odpovídají port pinům PTA1-PTA4. Dane piny jsem nakonfiguroval jako vstupní s připojením pull-up rezistorů. Dane piny se používají pro generování přerušení od modulu KBI. Piny ROW1-ROW4 jsem zapojil na piny A28, A42, B7 a B9 konektoru J9 a J8, které odpovídají port pinům PTA7, PTC1, PTB6, PTB7. Dane piny jsou použité pro postupně uzemnění řádků klávesnice.

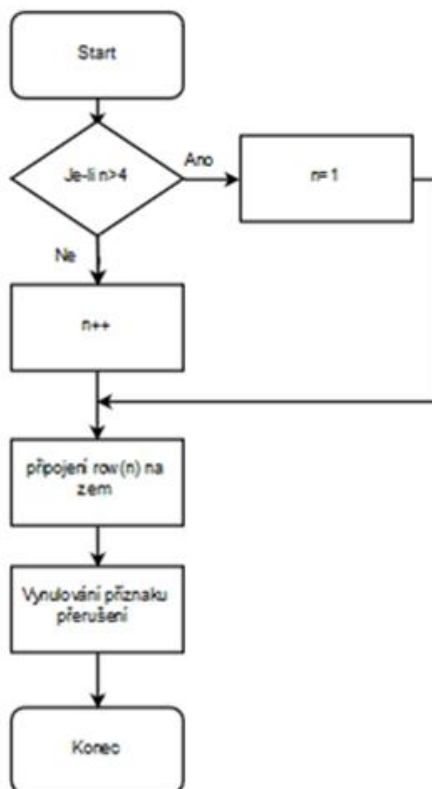


6. SOFTWARE PRO OVLÁDÁNÍ PERIFERIÍ

Demonstrační software je rozdělen na 4 části:

- 1) Soubory pro obsluhu modulu IIC mikrokontroleru.
- 2) Soubory pro obsluhu LCD displeje prostřednictvím IIC modulu
- 3) Soubory pro obsluhu kalendářového obvodu po IIC sběrnici
- 4) Hlavní program

V hlavním programu *main.c* jsou nadefinované 4 vektory přerušení: TPM1, TPM2, TOD a KBI. Přerušení od modulu TPM1 se používá pro postupné uzemnění řádků klávesnice. Vývojový diagram funkce přerušení od modulu TPM1 je na Obr.31. Modul TPM1 je nastaven na vygenerování přerušení každých cca 2 ms.



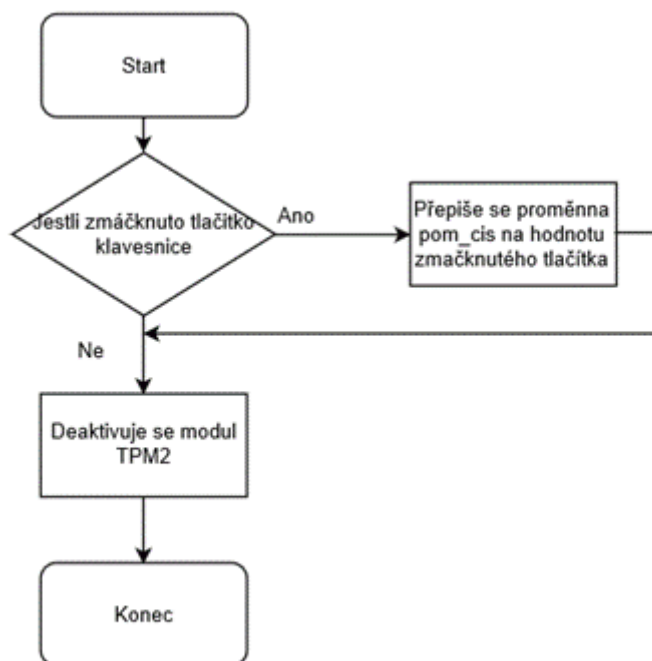
Obr. 29 Vývojový diagram funkce přerušení od modulu TPM1

Modul TPM2 se používá pro ošetření zámkitů tlačítek maticové klávesnice. Proměnná *pom_cis* se používá pro nastavení hodnot parametru kalendářového obvodu. Princip ošetření je následující:

- 1) Při zmáčknutí tlačítka vygeneruje se přerušení od modulu KBI, ve funkci kterého se zapne modul TPM2 a povolí se přerušení od modulu TPM2.

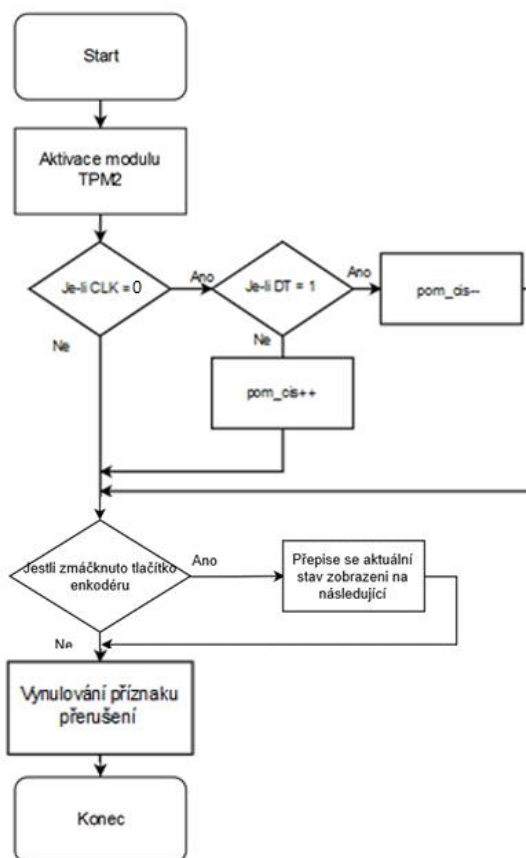
- 2) Modul TPM2 je inicializován tak, že přerušení se vygeneruje po cca 50μs od zapnutí modulu.
- 3) Ve funkci přerušení od modulu TPM2 se nachází kontrola stavu jednotlivých tlačítek kód jejich obsluhy.

Vývojový diagram je znázorněn na Obr.32



Obr. 30 Vývojový diagram funkce přerušení od modulu TPM2

Přerušení od modulu KBI se používá pro záznam stisknutí tlačítka enkodéru nebo maticové klávesnice. Vývojový diagram funkce přerušení od modulu KBI je na Obr. 33



Obr. 33 Vývojový diagram funkce přerušení od modulu KBI

Přerušení od modulu TOD se používá pro zobrazení veškeré potřebné informace na LCD displeji. Princip se spočívá v tom, že modul TOD každých 0,25s generuje přerušení. V přerušovací rutině se nachází kód obsluhy LCD displeje. Vývojový diagram je znázorněn na Obr. 34

Program vyplněn v podobě stavového automatu. Pomocí tlačítka enkodéru se přepínají stavy zobrazení údajů na displeji. Odeslání dat do kalendářového obvodu se při změně stavu.

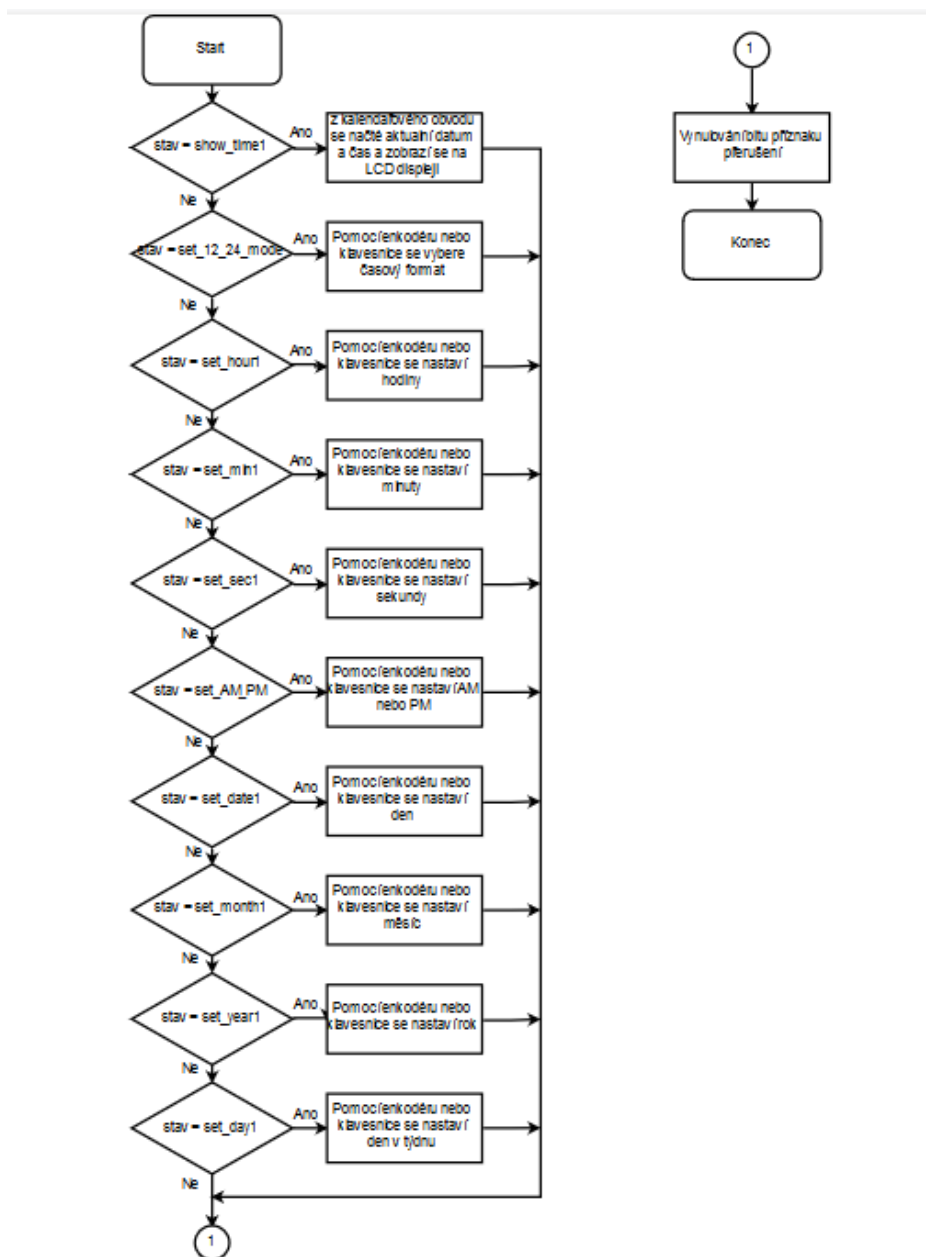
Soubory pro obsluhu IIC modulu obsahují funkce, potřebné pro činnost modulu IIC:

- *void iic_init (void)*. Funkce provádí počáteční inicializaci modulu IIC.

Nejprve modul je resetován pro smazání možných nastavených flagů, které mohou způsobit nekorektní činnost modulu. Dále se vyberou piny pro signály SDA a SCL a nastaví se baud rate na hodnotu ≈ 100 kbps, což odpovídá $MULT = 01$ ($mul = 02$) a $ICR = 0x16$ ($SCLdivisor = 104$) (zapiše se $0b01010110$ do registru IICF). Zakáže se přerušení.

Hodnoty bitů $MULT$ a ICR jsem zvolil tak, aby Baud rate byl přibližně roven ≈ 100 kbps.

$$IIC \text{ baud rate} = \frac{40 * 10^6}{2 * 2 * 104} = 96\,154 \text{ kbps};$$



Obr. 34 Vývojový diagram funkce přerušení od modulu TOD

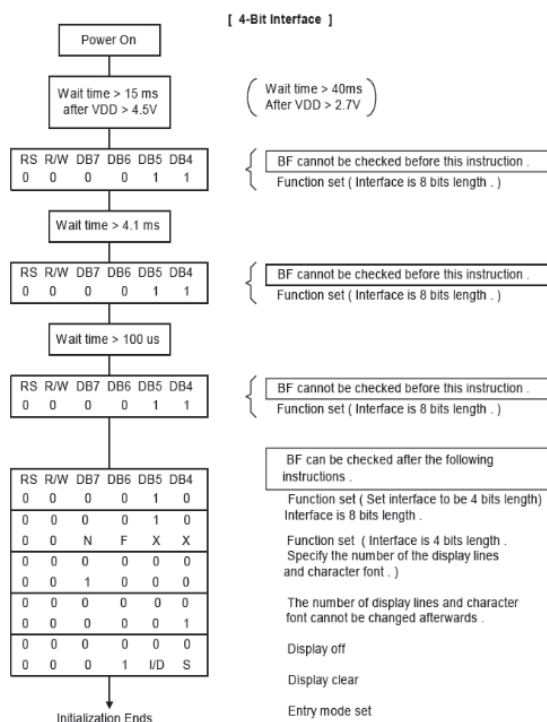
- *int iic_start (byte slave_address)*. Funkce pro generování Start signálu a následné odeslání adresy Slave zařízení doplněnou o RW bit. Na začátku modul je resetován pro smazání možných nastavených flagů. Dále vygeneruje se Start signál, vybere se směr komunikace a odešle se adresa Slave zařízení doplněna o RW bit. Zkontroluje se flag přerušení, který musí být nastaven odesláním 1 bajtu dat. Vynuluje se flag pro možnost

identifikaci dalších událostí. Funkce vrátí *Acknowledge* bit, signalizující úspěšné přijetí Slave zařízením 1 bajtu dat.

- *void iic_stop (void)*. Funkce generuje Stop signál.
- *int iic_send(byte data)*. Funkce pro odesílání 1 bajtu dat. Nejprve se změní směr komunikace. Odešle se bajt. Zkontroluje se flag přerušení. Vynuluje se flag. Funkce vrací *Acknowledge* bit.
- *int iic_repeat_start (byte slave_address)*. Funkce generuje Repeated Start signál. Následně odešle adresu Slave zařízení doplněnou o RW bit. Zkontroluje se flag přerušení. Vynuluje se flag. Funkce vrací *Acknowledge* bit.
- *byte iic_receive (void)*. Funkce provádí přijetí 1 bajtu dat od Slave zařízení. Nejprve se změní směr komunikace na přijetí dat. Zruší se odeslání *Acknowledge* bitu pro ukončení komunikace po odeslání 1 bajtu dat. Přečte se 1 bajt dat. Zkontroluje se flag přerušení. Vynuluje se flag přerušení. Změna směru komunikace.

Soubory pro obsluhu LCD displeje obsahují funkce, potřebné pro činnost LCD displeje:

- *void delay_us (int a)*. Funkce pro generování zpoždění, potřebné pro dodržování časování. Parametr *a* uvádí počet μ s.
- *void lcd_command (byte com)*. Funkce odesílá bajt *com* do instrukčního registru LCD displeje. Odeslání probíhá podle 4-bitovému interfejsu přes IIC převodník s dodržováním zpoždění signálů. Detailnější informace obsahuje kapitola 4.1.3
- *void lcd_init (void)*. Funkce pro počáteční inicializace LCD displeje. Inicializace se spočívá v odeslání několika komand do instrukčního registru podle inicializačního diagramu uvedeného v dokumentaci k displeji s dodržováním časování (Obr. 35).



Obr. 31 Inicializace LCD displeje

- *void lcd_show_char (char c).* Funkce pro odeslání znaku v ANSI kódu do datového registru LCD displeje. Funkce je velice podobná funkce *void lcd_command (byte com)* s tím rozdílem, že odeslání probíhá do datového registru. Po oddělení znaku do datového registru, znak automaticky zobrazí se na displeji.
- *void lcd_show_string (char *str).* Funkce pro odeslání několika znaku do datového registru displeje. Používá funkci *void lcd_show_char (char c).*
- *void lcd_show_hex (byte hex).* Funkce pro odeslání a zobrazení čísla podle BCD formátu. Používá se při zobrazení času a daty kalendářového obvodu.
- *void lcd_show_time (byte hour, byte min, byte sec, byte date, byte month, byte year, byte day).* Funkce pro odeslání a zobrazení času a daty na LCD displeji.

Soubory pro obsluhu kalendářového obvodu obsahují funkce, potřebné pro činnost RTC obvodu:

- *void rtc_set_sec(byte sec)*. Funkce pro nastavení hodnoty vteřin do kalendářového obvodu. Funkce využívá algoritmus popsany v kapitole 2.5.4.
- *void rtc_set_min(byte min)*. Funkce pro nastavení hodnoty minut do kalendářového obvodu.
- *void rtc_set_hour(byte hours)*. Funkce pro nastavení hodnoty hodin, hodinového formátu a příznaku AM/PM do kalendářového obvodu.
- *void rtc_set_date(byte date)*. Funkce pro nastavení dne do kalendářového obvodu.
- *void rtc_set_day(byte day)*. Funkce pro nastavení dne v týdnu do kalendářového obvodu.
- *void rtc_set_month(byte month)*. Funkce pro nastavení měsíce do kalendářového obvodu.
- *void rtc_set_year(byte year)*. Funkce pro nastavení roku do kalendářového obvodu.
- *byte rtc_read_sec(void)*. Funkce pro přečtení vteřin z kalendářového obvodu.
- *byte rtc_read_min(void)*. Funkce pro přečtení minut z kalendářového obvodu.
- *byte rtc_read_hour(void)*. Funkce pro přečtení hodin, hodinového formátu a příznaku AM/PM z kalendářového obvodu.
- *byte rtc_read_date(void)*. Funkce pro přečtení dne z kalendářového obvodu.
- *byte rtc_read_day(void)*. Funkce pro přečtení dne v týdnu z kalendářového obvodu.
- *byte rtc_read_month(void)*. Funkce pro přečtení měsíce z kalendářového obvodu.

- *byte rtc_read_year(void)*. Funkce pro přečtení roku z kalendářového obvodu.

7. NÁVRH LABORATORNÍCH ÚLOH

7.1 Laboratorní úloha č.1

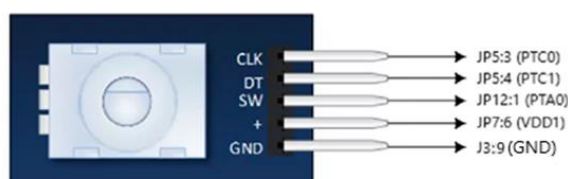
Cílem této laboratorní úlohy je seznámení studentů s problematikou připojení rotačního enkodéru ke vývojové desce TRW-S08LH64. Přepokládaná znalost studentů: studenti umí nakonfigurovat GPIO, ovládat LED diodami a tlačítkem. Přepokládaná časová náročnost: 3 akademické hodiny.

Zadaní:

1. Zapojte enkodér do desky TRW-S08LH64, správně nakonfigurujte příslušné porty mikrokontroleru. Na základě teoretického rozboru principu činnosti rotačního enkodéru, naimplementujte algoritmus pro obsluhu enkodéru tak aby při nízké úrovni signálu CLK se rozsvítila dioda LED1, při nízké úrovni signálu DT se rozsvítila dioda LED2 a při zmáčknutí tlačítka SW se rozsvítila dioda LED3
2. Naimplementujte algoritmus, který bude pomoci enkodéru inkrementovat nebo dekrementovat a pomoci diod LED1, LED2, LED3 a LED4 zobrazovat číslo ve dvojkové soustavě. Otáčení enkodéru vpravo číslo se inkrementuje o 1, vlevo dekrementuje se o 1. Pomocí LED 1 reprezentujte zde číslo je záporné nebo kladné tak, že se rozsvítí, jestli číslo je záporné. Pomocí zbývajících 3 diod zobrazte rozsah hodnot od -7 do 7. Stlačení tlačítka SW vynuluje zobrazovací číslo.
3. Upravte algoritmus tak, aby v přerušovací rutině TPM modulu se prováděla kontrola změny stavu signálu CLK a obsluha tlačítka SW.

Doporučený postup:

1. a) Zapojte piny podle obr. 36



Obr. 36 Připojení enkodéru k desce

- b) Nakonfigurujte příslušné porty tak, aby umožňovaly ovládaní diod LED1, LED2 a LED3.
 - c) Pin PTA0 nakonfigurujte jako vstupní pro čtení stavu tlačítka. Připojte pull-up rezistor. Piny PTC0 a PTC1 nakonfigurujte jako vstupní pro čtení stavu signálů CLK a DT. Pull-up rezistory připojovat není potřeba, protože na enkodéru už jsou připojeny.
 - d) V nekonečné smyčce hlavního programu čtete stav pinu PTC0 (CLK) a podle úrovně signálu, měňte stav diody LED1. Při nízké úrovni signálu CLK dioda LED1 se rozsvítí, při vysoké se zhasne. Princip je stejný pro pin PTC1 (DT) s tím, že řízená dioda je LED2. Zároveň je potřeba číst stav pinu PTA0 (SW) a při nízké úrovni tohoto signálu rozsvítit diodu LED3.
2. a) Vytvořte pomocnou proměnnou, kterou budete inkrementovat nebo dekrementovat. Pomocí teoretického rozboru principu činnosti rotačního enkodéru naimplementujte algoritmus pro vyhodnocení směru otáčení enkodéru. Realizujte inkrementace pomocné proměnné při otáčení enkodéru vpravo a dekrementace při otáčení enkodéru vlevo.
- b) Pomocí diod LED1, LED2, LED3 a LED4 zobrazte pomocnou proměnnou ve dvojkové soustavě tak, že LED1 bude reprezentovat znaménko čísla. Pokud LED1 je rozsvícena, to znamená že číslo reprezentující diodami LED2, LED3 a LED4 je záporné.
3. a) Aby nezatěžovat hlavní program kontrolou stavu pinu PTA0, PTC0 a PTC1 realizujte tuto kontrolu mimo hlavní program pomocí přerušení od modulu čítače TPM. Modul TPM nastavte tak, aby každých 1 ms vyvolá přerušení a zkontroluje stav pinů.

7.2 Laboratorní úloha č. 2

Cílem této laboratorní úlohy je seznámení studentů s modulem sériové sběrnice IIC a problematikou použití daného modulu pro připojení LCD displeje s IIC převodníkem. Přepokládaná časová náročnost: 6 akademických hodin.

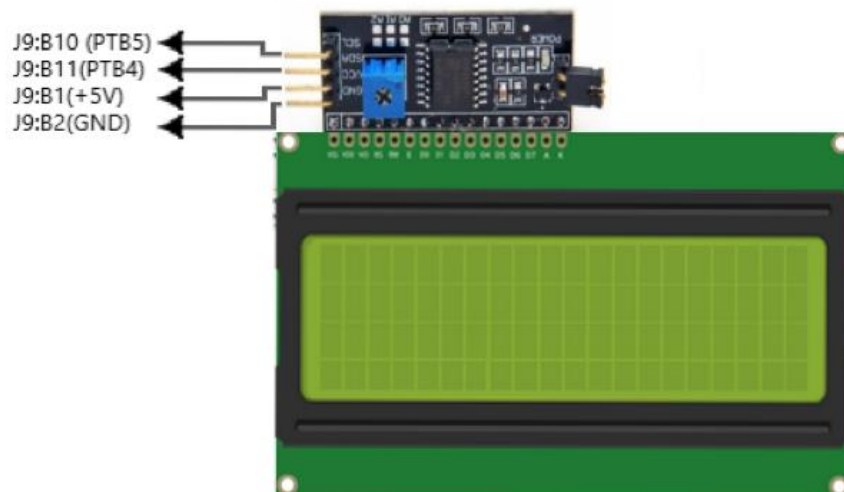
Zadání:

1. Seznámit se s principem činnosti sériové sběrnice IIC. Seznámit se s modulem IIC mikrokontroléru.

2. Zapojit LCD displej s převodníkem k vývojovému systému Freescale Tower System.
3. Napište funkci `void i2c_init (void)`, která provede počáteční inicializaci modulu IIC.
4. Napište funkci `void lcd_command (byte com)`, která odešle komandu do instrukčního registru LCD displeje přes IIC převodník.
5. Napište funkci `void lcd_init (void)` pro počáteční inicializaci LCD displeje
6. Napište funkci `void lcd_show_char (const char c)`, která odešle a zobrazí znak `c`.
7. Napište funkci `void lcd_show_string (const char *str)`, která postupně po znacích zobrazí řetězec na LCD displeji.

Doporučený postup:

1. a) Zapojit piny převodníku IIC podle Obr. 37



Obr. 37 Připojení LCD displeje s IIC převodníkem k Tower Freescale System

2. a) Napište funkci `void i2c_init (void)` pro inicializaci IIC modulu. Využijeme piny PTB4 pro SDA vodič, PTB5 pro SCL. Zakážeme přerušení. Baud rate nastavíme na 0b01010110.
3. a) Z teoretického rozboru zjistíte, že IIC převodník na bázi PCF8574 převádí 1 bajt dat z vodiče SDA na piny P0-P7 a naopak. Dále je potřeba zjistit, jak piny z IIC převodníku jsou připojené do LCD displeje.
b) Nadefinujte pomocnou proměnnou typu `byte pom` kterou budete odesílat do IIC převodníku. Je jasné že po odeslání proměnné `pom` do IIC převodníku první bit proměnné `pom` bude odpovídat bitu P0, který připojen na vývod RS, druhý bit – P1, který připojen na vývod RW, třetí bit – P2, připojený na vývod E,

čtvrtý bit – P3, připojený na vývod A, pátý až sedmý bity – P4 až 7, které připojené na vývod y D4-D7. Z toho vyplývá, že s displejem přes IIC převodník lze komunikovat pouze podle 4-bitovému interfejsu a pro odeslání komandy budeme využívat posledních 4 bitů pomocné proměnné *pom*.

c) Nejdříve proveďte inicializace pomocné proměnné *pom*. První bit, který odpovídá vývodu RS, nastavte na 0, protože budete zapisovat do instrukčního registru. Druhý bit 1, který odpovídá vývodu RW, nastavte na 0, protože budete chtít zapisovat. Třetí bit 2 (vývod E) nastavte na 0, protože zatím není potřeba posílat data do displeje, jen nachystat pro odesílání. Čtvrtý bit (vývod A) nastavte na 1, tím zapnete LED podsvícení displeje. Pátý až sedmý bity (vývody D4-D7) nastavte podle posledních 4 bitů komandy *com* (horní půlbajt), protože podle 4-bitovému interfejsu prvním se odesílá poslední půlbajt.

d) Pro začátek komunikace vygenerujte Start signál a následně odešlete adresu Slave zařízení doplněnou o RW bit, který v daném případě bude roven 0, protože budete chtít odesílat data Slave zařízení. Adresa IIC převodníku je 0x3F. Zkontrolujte flag přerušení. Jestli 1 bajt odeslán flag přerušení bude nastaven. Pro vymazání flagu přerušení je potřeba do něho zapsat hodnotu 1. Dále zkontrolujte, jestli byl přijat *Acknowledge* bit od Slave zařízení. Po úspěšném spojení se Slave zařízení můžete začít odesílat data do IIC převodníku. Pošlete dříve inicializovanou pomocnou proměnnou *pom*. Po každém odeslání 1 bajtu dat zkontrolujte flag přerušení, vynulujte flag a zkontrolujte přijetí *Acknowledge* bitu. Tím jste nachystali odesílání půlbajtu komandy do LCD displeje. Pro to, aby odeslat půlbajt komandy do instrukčního registru LCD displeje musíte nastavit vysokou úroveň na pin E. Proto třetí bit proměnné *pom*, odpovídající pinu E, nastavte na vysokou úroveň a odešlete na výstup IIC převodníku. Máte odeslaný půlbajt komandy. Teď musíte nastavit nízkou úroveň na vývod E, přepsat pátý až sedmý bity proměnné *pom* na zbyly půlbajtu komandy *com*, odeslat proměnnou *pom* na výstup IIC převodníku, nastavit vysokou úroveň na vývod E. Tím odešlete druhou polovinu bajtu komandy *com*. Nastavte vývod E na nízkou úroveň a znovu pošlete na výstup IIC převodníku. Pro ukončení komunikaci vygenerujte Stop signál. Komandu máte odeslanou.

4. a) Napište funkci *void lcd_init (void)* pro inicializaci displeje. Inicializace se spočívá v tom, aby v určitém pořadí a časových intervalech poslat komandy. Pořadí a časové intervaly jsou popsány v teoretickém rozboru.
5. a) Funkce pro zobrazení znaku je velice podobná s funkcí pro odeslání komandy s tím rozdílem, že odesílat budete znak do datového registru. Znak je kódován podle ANSI. Datový registr zvolíte tím, že RS pin nastavíte na vysokou úroveň.
6. a) Funkci pro zobrazení řetězce realizujte pomocí funkce pro zobrazení znaku. Postupně zobrazujte znaky z řetězce, pokud ne narazíte na znak konce řetězce.

8. ZÁVĚR

Bakalářská práce se zabývá připojením rozšiřujících periférií k vývojové desce TWR-S08LH64 a realizací dvou laboratorních úloh. Pro realizaci připojení periférií byl vytvořen přípravek na univerzálním pájivém poli. Na přípravku jsou umístěné kalendářový obvod, poskytující data o čase a datumu, LCD displej s IIC převodníkem, který zobrazuje poskytnutou informaci, maticová klávesnice a enkodér pro řízení a nastavení.

Dále jsem vytvořil tři knihovny, realizující obsluhu IIC modulu, LCD displeje 20x04 2004a a kalendářového obvodu DS3231. Ve hlavním souboru *main.c* jsou nadefinované 4 vektory přerušení pro obsluhu přípravku. Pro obsluhu enkodéru se používá přerušení od modulu KBI, pro obsluhu klávesnice se používají přerušení od modulů TPM1, KBI a TPM2, pro obsluhu LCD displeje se používá přerušení od modulu TOD, pro obsluhu kalendářového obvodu se používá přerušení od modulu KBI.

Na závěr jsem návrh dvě laboratorní úlohy se zadáním a doporučeným postupem demonstrující připojení enkodéru a LCD displeje k vývojové desce TWR-S08LH64. Úlohy byly odladěny a se nachází na přiloženém CD.

9. LITERATURA

- [1] Nxp.com. (2019). MC9S08LH 8-bit Segment LCD Tower System Module | NXP. [online] Available at: https://www.nxp.com/products/no-longer-manufactured/mc9s08lh-8-bit-segment-lcd-tower-system-module:TWR-S08LH64?fbclid=IwAR3ad4GoL18uEzDJD_2Y5vjrkqNPge6xu8p38ahw2Pt1udsN1zoTrFAT2wc [Accessed 5 May 2019].
- [2] Nxp.com. (2019). Tower System Configuration. [online] Available at: https://www.nxp.com/files-static/graphic/block_diagram/TWRSYSCONF_BD1.html?fbclid=IwAR3QEpzq-7ivp_EcHeghB_kdITc1-rayZZv7zgBhci3occHxNjiKNAq_h3w [Accessed 7 May 2019].
- [3] Nxp.com. (2019). [online] Available at: https://www.nxp.com/docs/en/reference-manual/MC9S08LH64RM.pdf?fbclid=IwAR1kArFEMrEXVpDzbBG2eQnygsGfQT32WpOstJZTDqB7nvZw8yp66EhII_A [Accessed 24 April 2019].
- [4] Интерфейсная шина ИС (I2C), (2019). [online] Available at: http://easyelectronics.ru/interface-bus-iic-i2c.html?fbclid=IwAR0LUq7ZdaDPV6wMdsMrCP_hPbvRhWKV6X3QrtB7-I48I0OWGPNQxnao05E [Accessed 10 May 2019].
- [5] Sparkfun.com. (2019). [online] Available at: https://www.sparkfun.com/datasheets/LCD/HD44780.pdf?fbclid=IwAR1vnp12IlhaMmJl-VcqsP__0Nf0ruP9dVmNCbgn9KbFUZ0L5gqcDxWLPv0 [Accessed 10 April 2019].
- [6] Electroschematics.com (2019). [online] Available at: https://www.electroschematics.com/12459/arduino-i2c-lcd-backpack-introductory-tutorial/?fbclid=IwAR0IRLrE0jKB8yKitJihnnGcmzp_kBsIdat5NY-sq7gGGl_sKZkrFh_QM40 [Accessed 7 April 2019].
- [7] Codius.ru. (2019). Инкрементальный энкодер: принцип действия, схемы подключения, работа с Arduino - Codius.ru. [online] Available at: http://codius.ru/articles/Инкрементальный_энкодер_принцип_действия_схемы

_подключения_работа_c_Arduino?fbclid=IwAR2IgXCgsTLNg6HDSyj8Ms_lu
QA2TMKyj6nfe2UPCIISZ0gpHYaamCuR38s [Accessed 8 May 2019].

- [8] GM electronic s.r.o, w. (2019). F-KV16KEY WHITE | GM electronic, spol. s.r.o.. [online] Gme.cz. Available at: <https://www.gme.cz/f-kv16key-white?fbclid=IwAR2irT-lYGYpdX4vFblwxAze7wTyaXzG5qnbgHrm3DBadCl7TDiyZjGifPk> [Accessed 10 May 2019].
- [9] Arduino8.webnode.cz. (2019). Lekce 23 - Arduino a klávesnice 4x4 :: Arduino. [online] Available at: https://arduino8.webnode.cz/news/lekce-23-arduino-a-klavesnice-4x3/?fbclid=IwAR1dtxMZGjD6v_QQSEZsFWJjydZKjr8VjcjNLOJgE68-gYaaAs6lnEh_6Wg [Accessed 10 April 2019].
- [10] GM electronic s.r.o, w. (2019). LCD I2C display 20x4 | GM electronic, spol. s.r.o.. [online] Gme.cz. Available at: <https://www.gme.cz/lcd-i2c-display-20x4?fbclid=IwAR1EtWf52JdpBySUocu4LTNiUS6UDFpn8GXdtCqXeEU5cYDu820tFEQnlb4#product-detail> [Accessed 10 April 2019].
- [11] GM electronic s.r.o, w. (2019). Modul Modul RTC DS3231, I2C a 32kb flash | GM electronic, spol. s.r.o.. [online] Gme.cz. Available at: https://www.gme.cz/modul-rtc-ds323-i2c-a-32kb-flash?fbclid=IwAR0IRLrE0jKB8yKitJihnnGcmzp_kBsIdat5NY-sq7gGGl_sKZkrFh_QM40 [Accessed 10 May 2019].
- [12] Datasheets.maximintegrated.com. (2019). [online] Available at: https://datasheets.maximintegrated.com/en/ds/DS3231.pdf?fbclid=IwAR00JvPs4MYQvt23xAF_12Mv5m_Jt2rNL47AkeCyyicdtp_9U8ivzv9zONg [Accessed 17 May 2019].

10. SEZNAM ZKRATEK

MSB	Most Significant Bit
LSB.....	Least Significant Bit
OSBDM.....	Open Source BDM
MCU.....	Micro Controller Unit
TPM.....	Timer Pulse-Width Modulator
TOD.....	Time of Day Module
LCD.....	Liquid Crystal Display

11. SEZNAM PŘÍLOH

ICR (hex)	SCL Divider	ICR (hex)	SCL Divider
00	20	20	160
01	22	21	192
02	24	22	224
03	26	23	256
04	28	24	288
05	30	25	320
06	34	26	384
07	40	27	480
08	28	28	320
09	32	29	384
0A	36	2A	448
0B	40	2B	512
0C	44	2C	576
0D	48	2D	640
0E	56	2E	768
0F	68	2F	960
10	48	30	640
11	56	31	768
12	64	32	896
13	72	33	1024
14	80	34	1152
15	88	35	1280
16	104	36	1536
17	128	37	1920
18	80	38	1280
19	96	39	1536
1A	112	3A	1792
1B	128	3B	2048
1C	144	3C	2304
1D	160	3D	2560
1E	192	3E	3072
1F	240	3F	3840

Tab. 2 Závislost SCLdivider na bitech ICR